FIG. 1

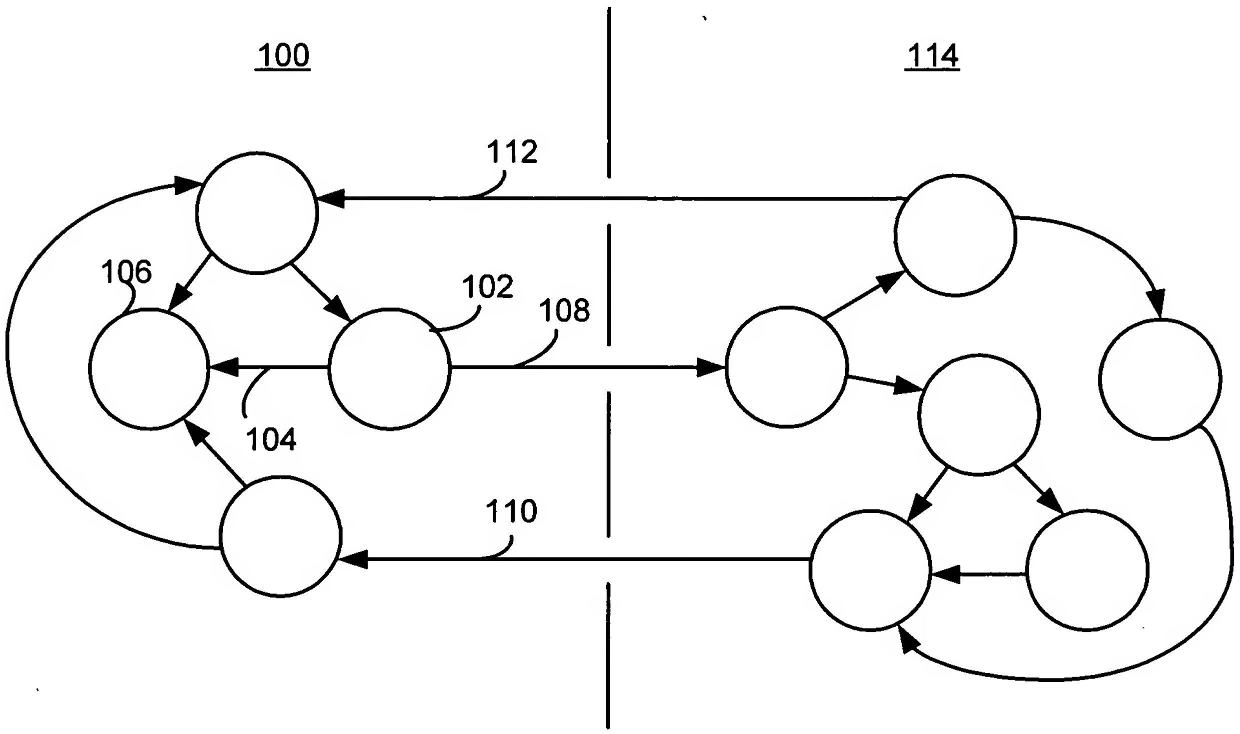


FIG. 2

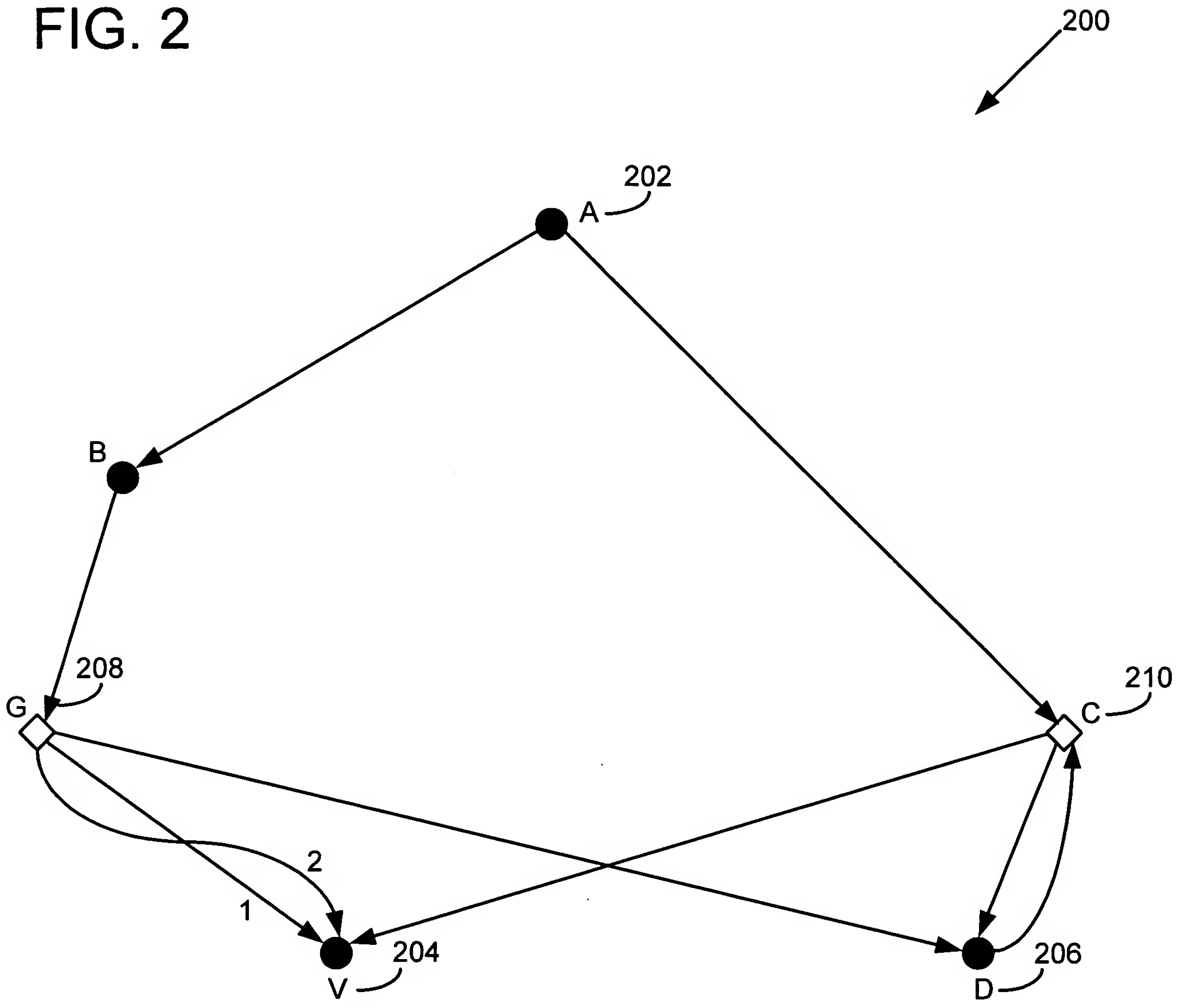


FIG. 3

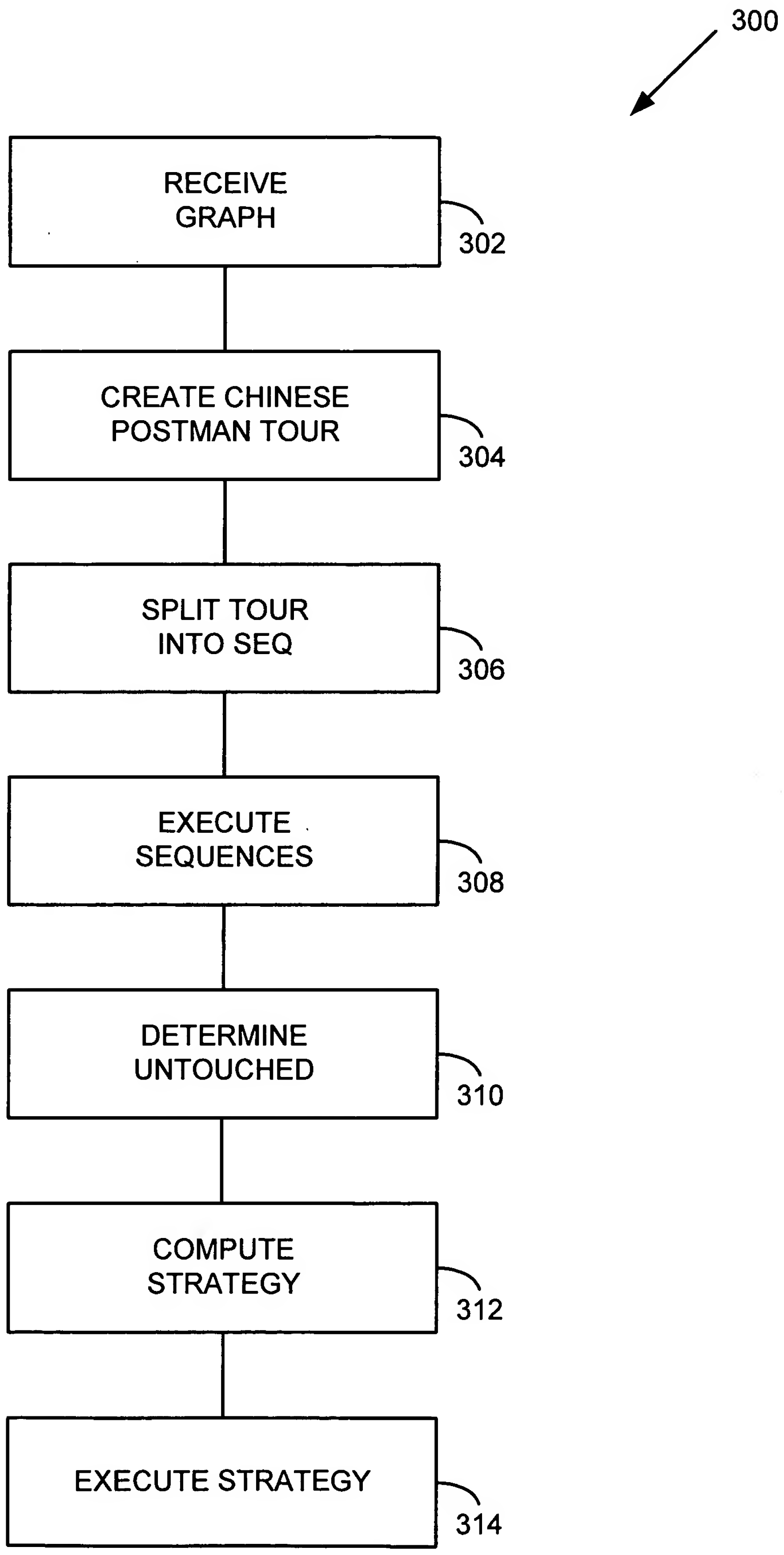

300
RECEIVE GRAPH
302
CREATE CHINESE POSTMAN TOUR
304
SPLIT TOUR INTO SEQ
306
EXECUTE SEQUENCES
308
DETERMINE UNTOUCHED
310
COMPUTE STRATEGY
312
EXECUTE STRATEGY
314

FIG. 4

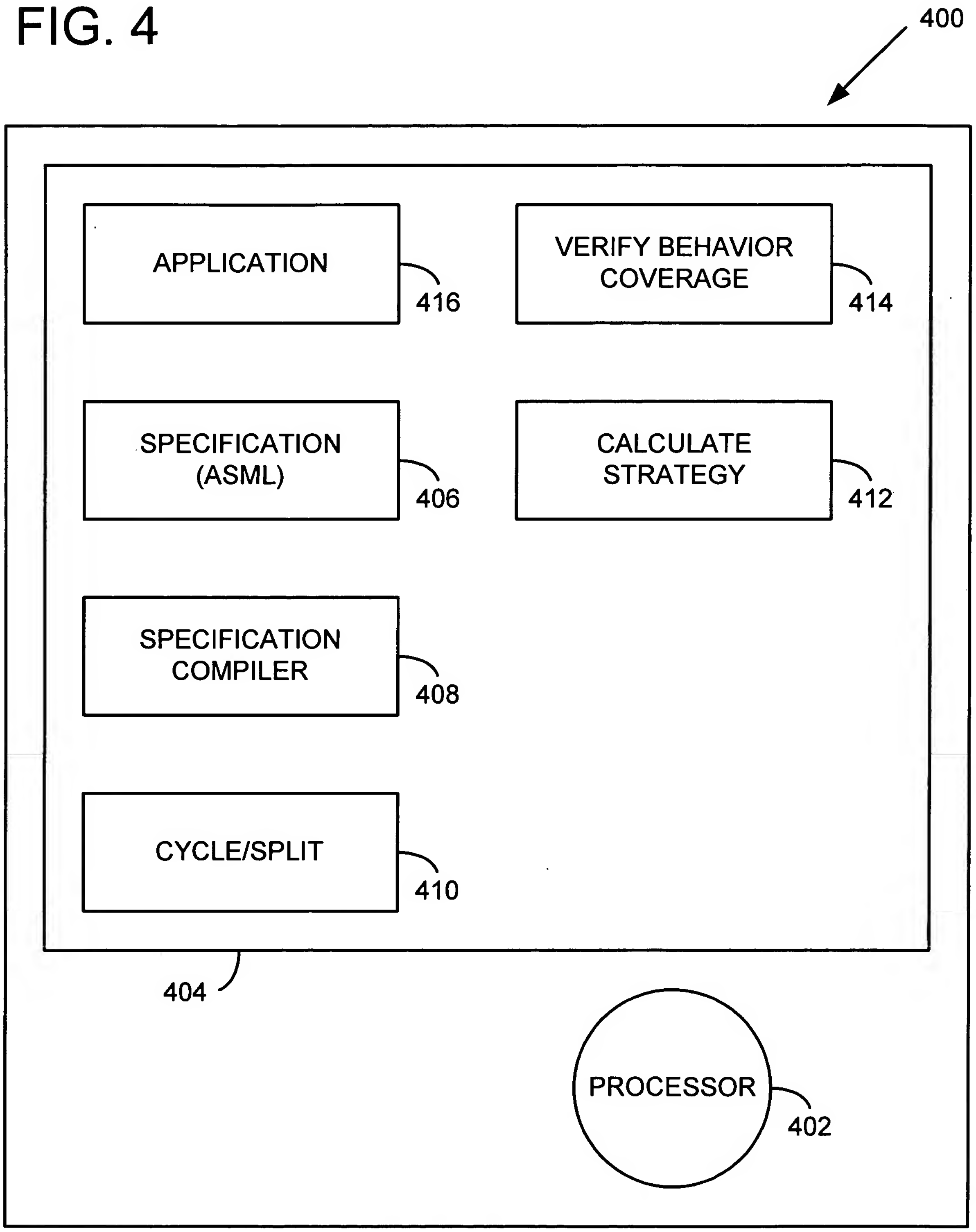

400
APPLICATION
416
VERIFY BEHAVIOR COVERAGE
414
SPECIFICATION (ASML)
406
CALCULATE STRATEGY
412
SPECIFICATION COMPILER
408
CYCLE/SPLIT
410
404
PROCESSOR
402

FIG. 5

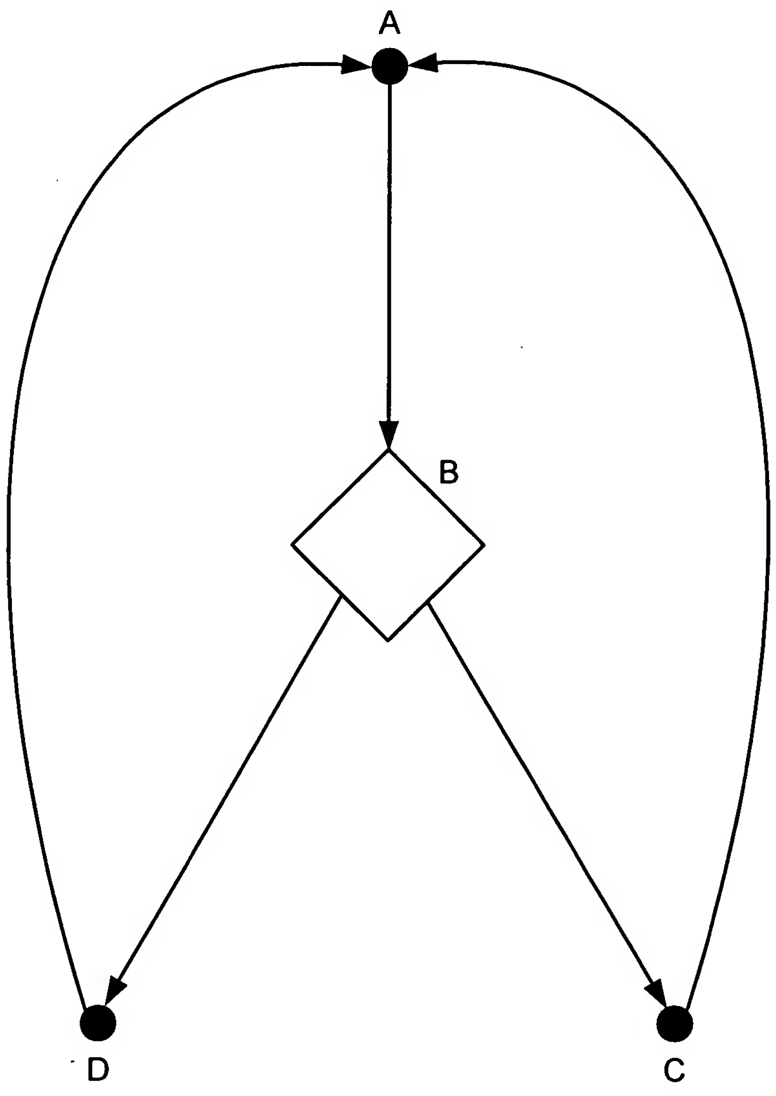


FIG. 6

600

```
    Initilalize() {
602   front:=P // P is victory set
604   newfront:={}
606
608   foreach v in P // initialize vertices of victory set
610     foreach i from 0 to n { // n is number of maximum edges allowed
612       Pr(v,i):=1 // vertices of victory set have probability of 1
614       C(v,i):=0 // no edge costs since this is victory set
616       S(v,i):=null // no strategies leave the victory set
618     }

620   foreach v in V-P {
622       for each i from 0 to n {
624       Pr(v,i):=0 // all other vertices initialized with zero probability,
626       C(v,i):=0 //                                  cost and strategy
628       S(v,i):=null //
        }
    }
```

FIG. 7

VERTEX	PROB COST STRATEGY	N = 0... 4				
		0	1	2	3	4
V	PR	1	1	1	1	1
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL
A	PR	0	0	0	0	0
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL
B	PR	0	0	0	0	0
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL
G	PR	0	0	0	0	0
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL
C	PR	0	0	0	0	0
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL
D	PR	0	0	0	0	0
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL

FIG. 8

800

```
802  StrategyCalculation(n){
804    for (i=1; i<=n; i++){
806      foreach v in front
808        Process(v,i)
810      foreach v in newfront
812        foreach k:=i+1;k<=n;k++ {
814          S(v,k):=S(v,i)
816          Pr(v,k):=Pr(v,i)
818          C(v,k):=C(v,i)
           }

820      Visited+=newfront
822      front:=newfront
824      newfront:={}

     }
   }
```

FIG. 9

900

```
902  Process(vertex v,i){
904    foreach edge entering v {
906      let u=edge.source // u is the source vertex of edge
908      if u is not in P // never exit from P, the victory set
910          if u is deterministic
912            if ImprovingOnEdge( edge,i){
914              S(u,i):=edge
916              Pr(u,i):=Pr(v,i-1)
918              C(u,i):=cost(edge)+C(v, i-1)
920              newfront:=newfront U {u}  // add u to newfront
               }
922          else  {//the node edge.source is nondeterministic

924              oldPr=Pr(u,i)
926              oldC= C(u,i)
928              if i>1 then
930                Pr(u,i)+=p(edge)(Pr(v,i-1)-Pr(v,i-2))
                 else
932                Pr(u,i)+=p(edge)Pr(v,i-1)

934              C(u,i) = max{cost(e)+ C(e.target,i-1): e exits from u}

936              if( oldPr ≠ Pr(u,i) or oldC ≠ C(u,i))
938                newfront:=newfront U {u}
             }
     }
   }

940  bool ImprovingOnEdge(edge, v, i)
       return
942    (Pr(edge.target,i-1),cost(edge)+ C(edge.target,i-1)) <
       (Pr(edge.source,i),C(edge.source,i))
```

FIG. 10

1000

VERTEX	PROB COST STRATEGY	N = 0... 4				
		0	1	2	3	4
V	PR	1	1	1	1	1
	C	0	0	0	0	0
	S	NULL	NULL	NULL	NULL	NULL
A	PR	0	0	1/2	2/3	3/4
	C	0	0	2	3	4
	S	NULL	NULL	AC	AB	AC
B	PR	0	0	2/3	2/3	2/3
	C	0	0	2	2	2
	S	NULL	NULL	BG	BG	BG
G	PR	0	2/3	2/3	2/3	2/3
	C	0	1	1	1	1
	S	NULL	NULL	NULL	NULL	NULL
C	PR	0	1/2	1/2	1/2	1/2
	C	0	1	1	1	1
	S	NULL	NULL	NULL	NULL	NULL
D	PR	0	0	1/2	1/2	1/2
	C	0	0	2	2	2
	S	NULL	NULL	DC	NULL	NULL

FIG. 11

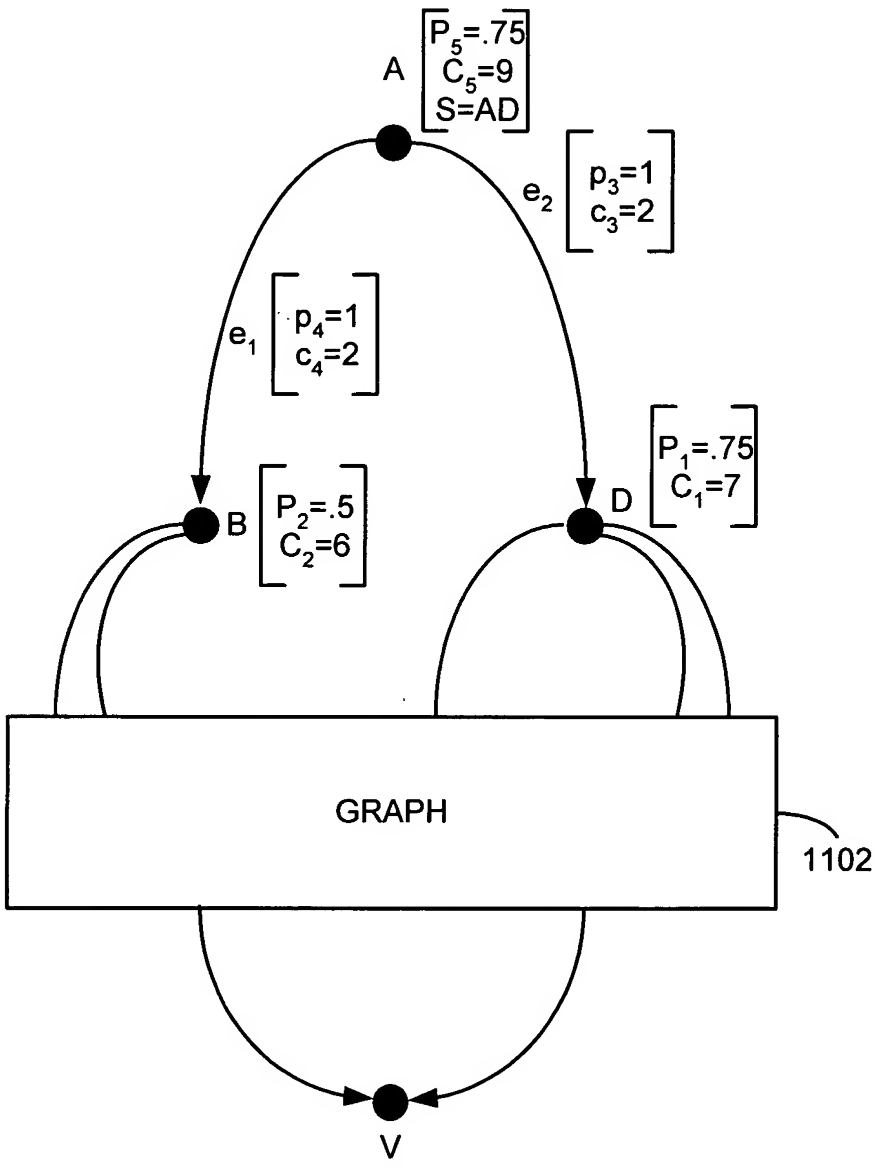

A
$P_5=.75$
$C_5=9$
$S=AD$
e_2
$p_3=1$
$c_3=2$
e_1
$p_4=1$
$c_4=2$
$P_1=.75$
$C_1=7$
D
B
$P_2=.5$
$C_2=6$
GRAPH
1102
V

FIG. 12

```
Traverse(node v,integer k){

  while (k>0){
     k:=k-1
     let e=edge choosen by the environment

     Ti:=choose any Ti starting with e
     cover every edge of Ti
     v:=end of Ti
  }
}
```

FIG. 13

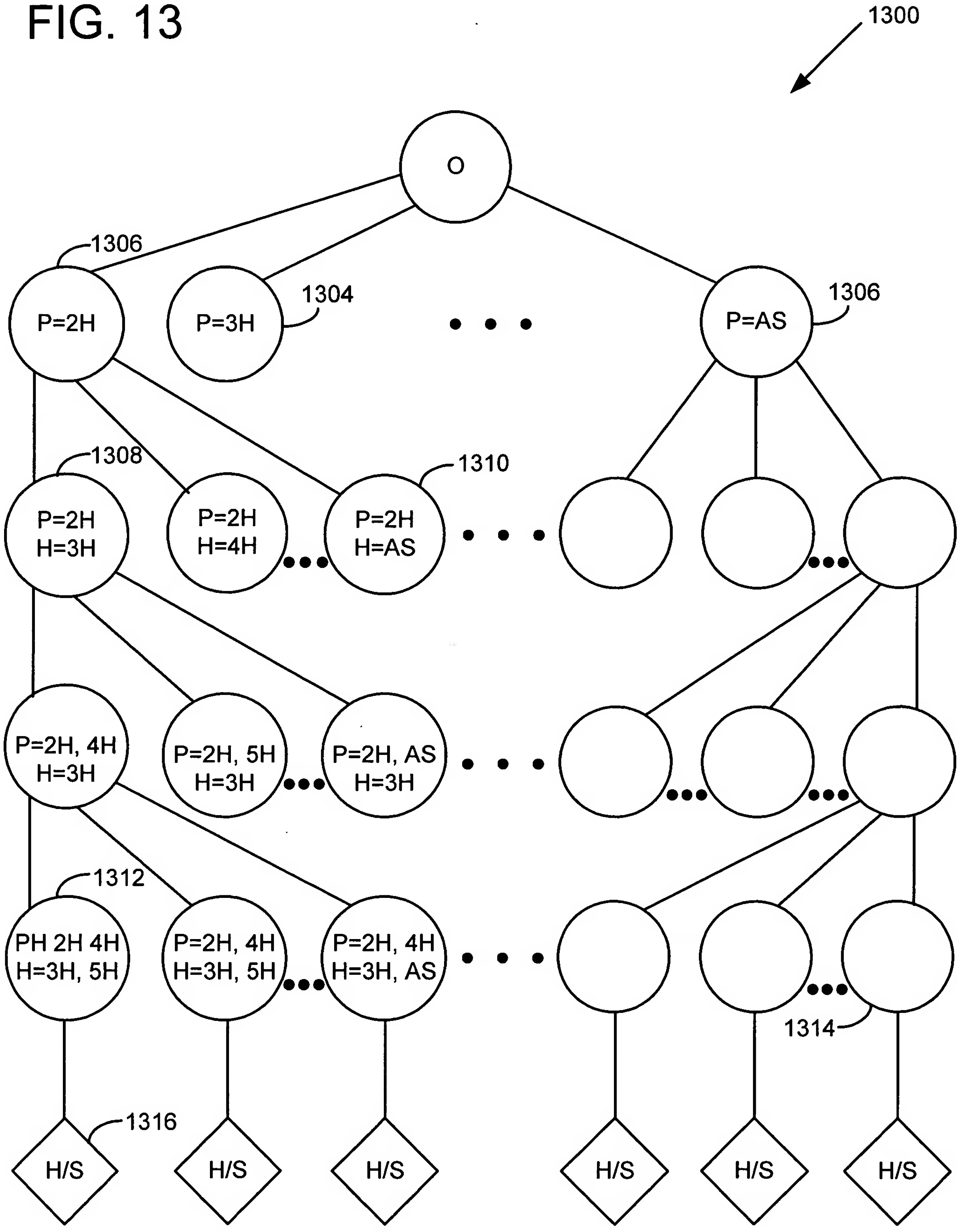
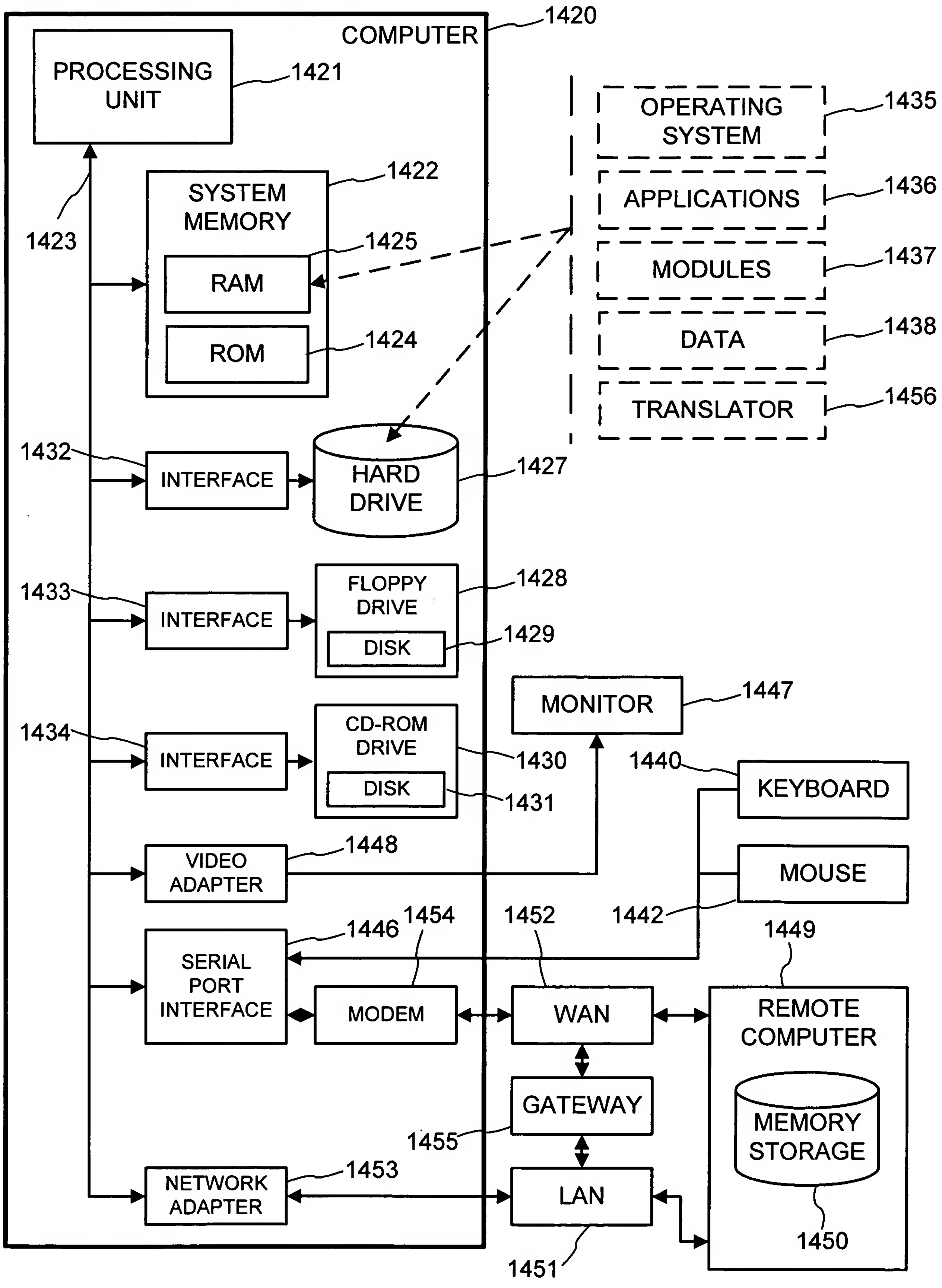

1300
O
1306
P=2H
P=3H
1304
P=AS
1306
1308
P=2H H=3H
P=2H H=4H
1310
P=2H H=AS
P=2H, 4H H=3H
P=2H, 5H H=3H
P=2H, AS H=3H
1312
PH 2H 4H H=3H, 5H
P=2H, 4H H=3H, 5H
P=2H, 4H H=3H, AS
1314
1316
H/S
H/S
H/S
H/S
H/S
H/S

FIG. 14


COMPUTER
1420
PROCESSING UNIT
1421
1423
SYSTEM MEMORY
1422
RAM
1425
ROM
1424
OPERATING SYSTEM
1435
APPLICATIONS
1436
MODULES
1437
DATA
1438
TRANSLATOR
1456
1432
INTERFACE
HARD DRIVE
1427
1433
INTERFACE
FLOPPY DRIVE
1428
DISK
1429
1434
INTERFACE
CD-ROM DRIVE
1430
DISK
1431
MONITOR
1447
1440
KEYBOARD
MOUSE
VIDEO ADAPTER
1448
1446
1454
1452
1442
1449
SERIAL PORT INTERFACE
MODEM
WAN
REMOTE COMPUTER
MEMORY STORAGE
1450
GATEWAY
1455
NETWORK ADAPTER
1453
LAN
1451